

# Graph Theory

Heena Jand<sup>a\*</sup>, Jasveer Kaur<sup>b\*</sup>

<sup>a,b</sup>CSE Deptt, Devki Devi Jain College for Women, Ludhiana-141007, India

<sup>a</sup>Email: [aheenajand21@gmail.com](mailto:aheenajand21@gmail.com)

<sup>b</sup>Email: [Jasveer.kaur133@gmail.com](mailto:Jasveer.kaur133@gmail.com)

## Abstract

Graph Theory is a graphical representation of a set of vertices which are connected by edges and is basically suitable in computer science and mathematics to create structural models. This paper describes the basic terminology of graphs, different types and connectivity of graphs to lead new inventions and modifications in the existing environment for enhancement in these fields.

**Keywords:** Vertices; Edges; Special Graphs; Connectivity of Graphs.

## 1. Introduction

Graphs, Directed Graphs and Trees appear in many areas of Mathematics and Computer Science. Graph is a non linear data structure. It is a widely used in applications like graph of any location, strategic problem etc. Graphs are basically designed and developed to store and analyze metadata and the connections between data. A graph  $G$  is defined as a set of two things which are  $G = (V, E)$ , where  $V$  stands for set of vertices of  $G$  and  $E$  stands for set of edges of  $G$ . There exists a mapping from the set of edges to a set of pairs of elements of  $V$ . An edge between  $u$  and  $v$  is denoted as  $e(u,v)$ .

### 1.1. Directed Graph

A directed graph  $G$  is also called a digraph or graph which consists of a non empty set of vertices  $v$  and a set of directed edges  $E$ .

---

\* Corresponding author.

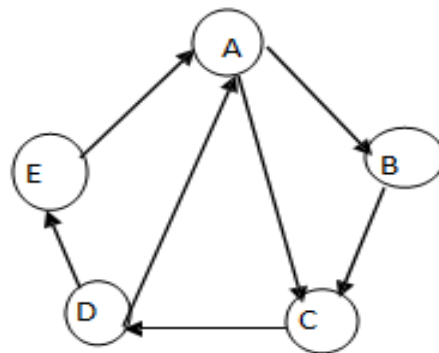
Each edge  $e$  in  $G$  is assigned a direction and identified with an ordered pair  $(u,v)$  of nodes in  $G$  where  $u$  represents the origin point of  $e$  and  $v$  represents destination point of  $e$ . Basic terminology used in Directed Graph  $G$  is:

$e$  starts at  $u$  and ends at  $v$

$u$  is the source point and  $v$  is the destination point

$u$  is the predecessor of  $v$  and  $v$  is the successor of  $u$ .

both are adjacent to each other

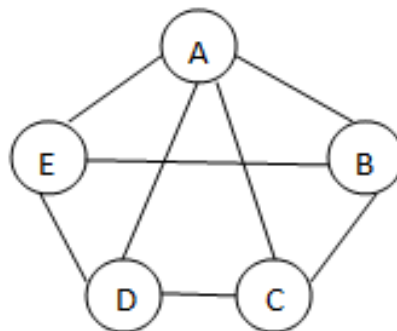


**Figure 1:** Representation of Directed Graph

### 1.2 Undirected Graph

An undirected graph is defined as a set of vertices that are connected together where all the edges are bi directional. An Undirected Graph  $G$  is also called a unidirectional network which consists of the set  $V$  of vertices and set  $E$  of edges which are unordered pairs  $(u,v)$  of elements of  $V$ .

Here both  $u$  and  $v$  can be act as a starting or ending points. Both must be adjacent to each other.

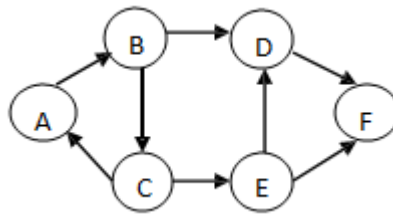


**Figure 2:** Representation of Undirected Graph

## 2. Basic Terminologies of Graph

### 2.1 Vertices and Edges

A vertex or node is the fundamental unit of which graphs are formed. A vertex is usually represented by a circle with label and an edge is represented by a line or arrow extending from one vertex to another in a graph. The two vertices forming an edge are said to be the endpoints of this edge and the edge is said to be incident of the vertices.



**Figure 3**

**Table 1:** Represents degrees of vertices

Vertex	Indegree	Outdegree
A	1	1
B	1	2
C	1	2
D	3	0
E	0	3
F	1	1

### 2.2 Degree

The degree of a vertex is the number of edges ends at that vertex. Degree of a directed graph is divided into two categories:

#### 2.3 In degree

In-degree of a vertex  $V$  is the number of edges  $e$  with  $v$  as their destination vertex.

#### 2.4 Out degree

Out-degree of a vertex  $V$  is the number of edges  $e$  with  $v$  as their origin vertex.

### 2.5 Edges in Series

When two edges in a graph have exactly one vertex in common and their vertex is of degree two, then two edges are said to be edges in series.

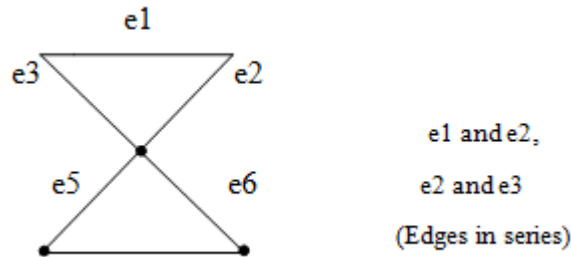


Figure 4

### 2.6 Multiedges

Distinct edges  $e$  and  $e'$  are called Multiedges if they connect the same end points i.e. if  $e=(u,v)$  and  $e'=(u,v)$

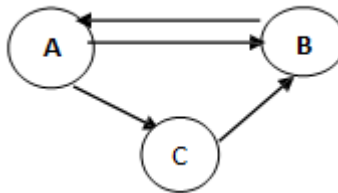


Figure 5

### 2.7 Self-loop

An edge  $e$  is said to be a loop if it has same starting and ending point. Loops are not allowed in directed graph.

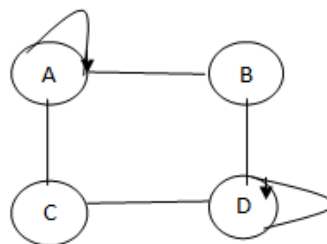


Figure 6

## **2.8 Order of Graph**

In a graph  $G$ , the number of vertices denoted by  $|V(G)|$  is called order of  $G$ .

## **3. Connectivity of Graph**

### **3.1 Walk**

A walk is called edge Train or chain. A walk in a graph  $G$  is finite sequence of vertices.  $W = V_0 e_1, V_1 e_2, \dots, V_{k-1} e_{k-1}, V_k$

#### **3.1 a. Walk is of two types**

*Open walk:* An open walk is an walk in which it begins and ends with different vertices

*Closed walk:* In closed walk, initial and terminal vertices of a graph are same. A walk contain no edge and has length zero is called Trivial Walk.

### **3.2 Path**

An open walk in which no vertex appears more than once is called path or simple path. Path doesn't intersect itself.

### **3.3 Simple path**

If all the vertices are distinct then it is simple path.

### **3.4 Trial**

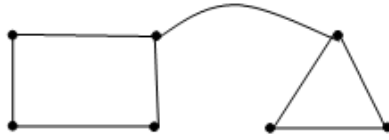
In this all the edges are distinct.

### **3.5 Length of Path**

Number of edges appearing in the sequence of a path is known as length of path. An edge which is not a self loop is a path of length 1. A self loop can be included in a walk but not in path.

### **3.6 Bridge**

Let  $G$  be a connected graph. Then an edge  $e$  is called bridge if by deleting  $e$ ,  $G$  become disconnected or we can say  $G-e$  is disconnected bridge is also called cut edge.



**Figure 7**

### 3.7 Cut vertex

A cut vertex is a vertex that when removed (with its boundary edges) from a graph creates more components than previously in the graph.

### 3.8 Circuit

Circuit is a closed path in which no vertex (except initial and terminal) appears more than once. It is also called cycle, elementary cycle or circular path or polygon. A self loop is a circuit but converse is not true.

The degree of every vertex in a circuit is two.

## 4. There are different types of Graphs which we learn in the following section

### 4.1 Null Graph

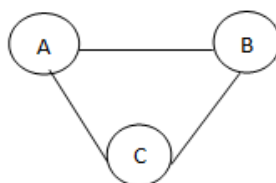
A graph that has no edges is said to be Null Graph and its vertices are denoted by  $N_n$ , where  $n$  represents number of vertices.



**Figure 8:** Representation of Null Graph

### 4.2 Simple Graph

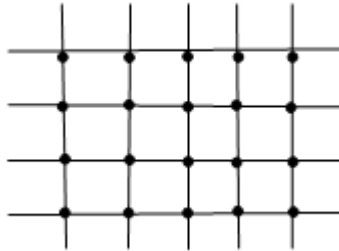
A Graph is said to be simple graph which contains neither loop non Parallel edges.



**Figure 9:** Representation of Simple Graph

### 4.3 Finite Graph

If a set of vertices  $V$  in graph  $G$  is a finite set then it is called finite graph.



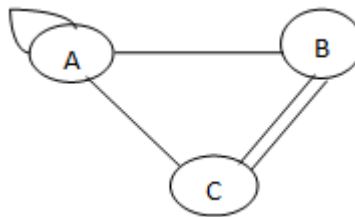
**Figure 10:** Representation of Infinite Graph

### 4.4 Infinite Graph

If a set of vertices of  $V$  in graph  $G$  is an infinite set then it is called infinite graph.

### 4.5 Multigraph

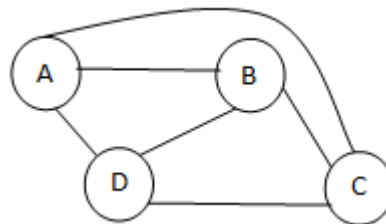
A Graph is said to be multigraph which has either loop or parallel edges or both.



**Figure 11:** Representation of Multigraph

### 4.6 Regular Graph

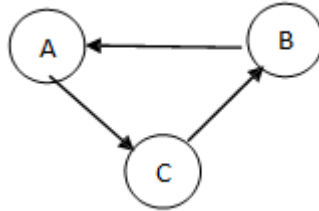
A graph is regular if all the vertices of the graph have the same degree. A graph is  $K$ -Regular Graph if all the vertices have same degree  $K$ .



**Figure 12:** Representation of Regular Graph

#### 4.7 Cycle Graph

If a graph consists of a single cycle, it is called cycle graph. The cycle graph with  $n$  vertices is denoted by  $C_n$ .



**Figure 13:** Representation of Cycle Graph

#### 4.8 Acyclic Graph

A graph  $G$  with no cycle is said to be Acyclic Graph.

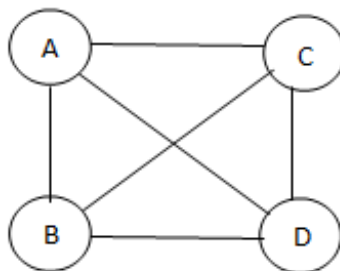
### 5. Some Special Graphs

In this section we will introduce several classes of special graphs which are widely used as examples and arise in many applications.

#### 5.1 Isomorphic Graph

We often need to know whether it is possible to draw two graphs in the same manner. For example in chemistry graphs are used to model compounds. Different compounds can have the same molecular formula but can differ in structure. Isomorphism is a useful terminology for graphs in which the word “isomorphism” comes from the Greek word ‘isos’ for equal and ‘morpha’ for form. So, two simple graphs are isomorphic if there is a one-to-one relationship between vertices of the two graphs.

In other words, we can say that two graphs are isomorphic if we can label both graphs with same labels or vertices so that every vertex has exactly the same neighbors in both graphs.



**Figure 14:** Representation of Isomorphic Graph



### 5.2 Homomorphic Graph

Two graphs are said to be homomorphic if they can both be obtained from a common graph by a sequence of replacing edges by simple chains. In other words, any graph G, obtain new graph H by dividing an edge of G with additional vertices.

$h:GH$  such that  $-(x,y) \in E(G) \implies (h(x),h(y)) \in E(H)$

It maps adjacent vertices of graph G to the adjacent vertices of the graph H. A homomorphism is an isomorphism if it is a bijective mapping.



Figure 15: Representation of Homomorphic Graph

### 4.3 Complete Graph

A complete graph is also known as Universal Graph. If a graph with n vertices in which each vertex is connected to each of the others (with one edge between each pair of vertices). The complete graph with n vertices is denoted by  $K_n$ .

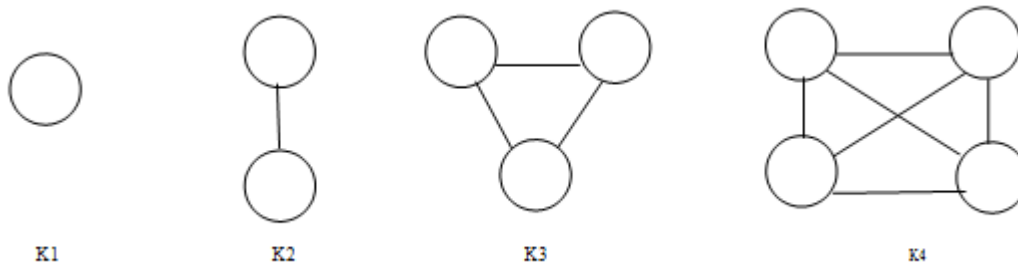
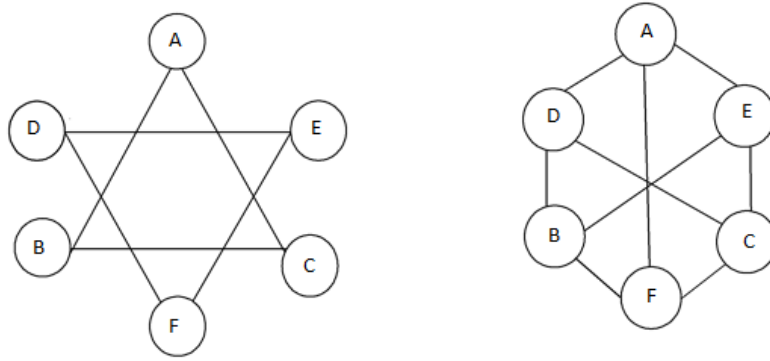


Figure 16: Representation of Complete Graph

### 5.4 Complement of a graph

In graph G, to find the complement of a graph you fill in all the missing edges to get a complete graph and remove all the edges those are already there. In this, it is not the set complement of the graph only the edges are complemented. It is also called Inverse of the graph.



**Figure 17:** Representation Complement of Graph

**5.5 Bipartite Graph**

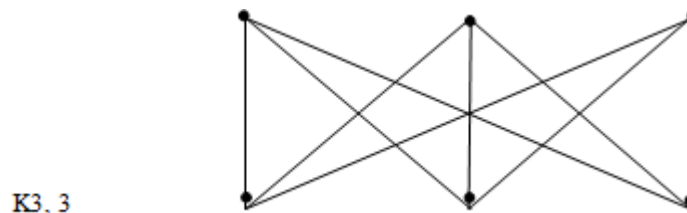
A bipartite graph, also called a bi-graph is a set of graph vertices decomposed into two disjoint subset A and B such that every edge in G joins a vertex in A with vertex B then graph is said to be bipartite graph.

This graph is denoted by  $K_{mn}$  where m is the number of vertices in A and n is the number of vertices in B. Trees are the example of bipartite graphs. A bipartite graph does not contain any odd length cycles.



**Figure 18:** Representation of Bipartite Graph

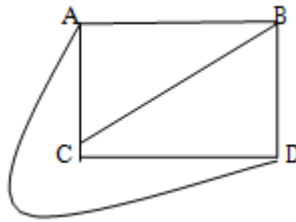
*Complete Bipartite Graph:* Bipartite graph is said to be complete bipartite graph if every vertex in A is joined to every vertex in B. It is denoted by  $K_{mn}$  where m and n are number of vertices in set A and B.



**Figure 19:** Representation of Complete Bipartite Graph

**5.6 Planar and Non Planar Graph**

A graph or multigraph which can be drawn in the plane without crossing any other edge is called planar graph. In other words one can say, A graph is a planar graph in which no two edges intersect each other.



**Figure 20:** Representation K4 Planar Graph

Whereas, Non planar graph is that cannot be represented on paper without crossing other edges.



**Figure 21:** Representation of Non Planar Graph

### 5.7 Connected and Disconnected Graphs

An undirected graph is called connected graph if there is a path between every pair of vertices in graph G. *Strongly Connected Graph:* A directed graph is strongly connected if there is a path from a to b and from b to a whenever a and b are vertices in the graph. *Unilaterally Connected Graph:* A simple directed graph is said to be unilaterally connected if for any pair of vertices of the graph, at least one of the vertices of the pair is reachable from other vertex. *Weakly Connected Graph:* A directed graph is weakly connected if there is a path between every two vertices in the undirected graph.

### 5.8 Hamiltonian Graph

Hamiltonian graph if it admits a Hamiltonian cycle is called Hamiltonian Graph. Hamiltonian graph is also called as Hamilton graph. Hamiltonian Circuit visits each vertex exactly once except for the first vertex, which is also last. A path passing through all the vertices of a graph is called a Hamiltonian path and a graph containing a Hamiltonian path is said to be traceable. Hamiltonian Path does not include any self loop or parallel edge.

If the last edge of a Hamiltonian cycle is dropped we get a Hamiltonian path. However, a non Hamiltonian graph can have a Hamiltonian path that is Hamiltonian path cannot always be used to form Hamiltonian cycle. Let G be a connected graph with n vertices  $n > 2$ . Let u and v are two non adjacent vertices in G and  $\deg(u) + \deg(v) \geq n$  then G is Hamiltonian. A graph can have both Hamiltonian path as well as Hamiltonian cycle. If a graph has

Hamiltonian cycle then it also has Hamiltonian path but converse is not true. If  $m$  be number of edges in  $G$ , if  $m \geq \frac{1}{2}(n^2 - 3n + 2)$ . Where  $n$  is number of vertices of  $G$  then graph is Hamiltonian graph.

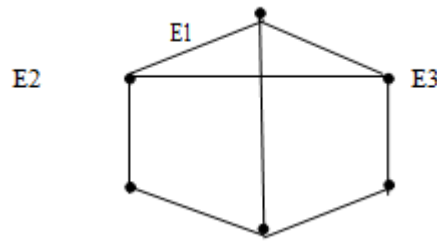


Figure 22

### 5.8 Euler Graph

*Euler Path:* A simple path in a graph  $G$  is called Euler path if it traverse every edge of graph exactly once. *Euler Circuit:*

Euler Circuit is a circuit in a graph  $G$  which traverses every edge of graph exactly once. Euler circuit is simply a closed Euler path. It is also called Euler Line.

*Euler Graph:* A closed walk in a graph  $G$  containing all the edges of  $G$  is called an Euler line in  $G$ . A graph containing an euler line is called an euler graph or one can say that, a graph which contain either Euler path or euler cycle is called euler graph. A connected graph  $G$  is a euler graph if all the vertices of  $G$  are of even degree. Its converse is possible.

If a graph has euler graph then it has either no vertex of odd degree or two vertices of odd degree. its converse is possible.

### 5.9 Weighted Graph

Weighted graphs can be directed or undirected. A graph in which we associate some cost or weight to the traversal of an edge is called weighted graph.

An example of weighted graph in the road network weight of each road may be its length or minimal time needed to drive. Weighted graph are used to model computer network, communication costs, the response time of the computer over these lines or the dist between computers, can all be studied using weighted graph.

### 6. Graph Coloring

A graph  $G$ , a vertex coloring of  $G$  is an assignment of colors to the vertices of  $G$  such that adjacent vertices have different colors. The minimum number of colors needed to paint  $G$  is called the chromatic number of  $G$  and is denoted by  $X(a)$ .

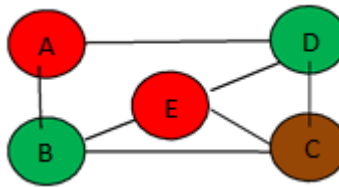
*Algorithm:*

Step 1: Color first vertex n1 with color c1.

Step 2: Do following for remaining n-1 vertices:

- a) Assign c1 to each vertex which is not adjacent to a previous vertex which was assigned c1.
- b) Repeat (a) with second color c2 and the subsequence of non colored vertices.
- c) Repeat (a) with third color c3 until all the vertices are colored.

Step 3: Exit.



**Figure 23**

*Applications of Graphs :*

Job Assignments

Local Area Network

Road Maps

Inter Connection Network for Parallel Computation

Robot Planning

Neural Networks

## **7. Conclusion**

This paper includes necessary initial set of definitions and concepts, along with examples. The main objective of this paper is to clear the importance of graph theory in various fields to solve complex problems such as mapping, program analysis, routing and switches in networking field etc.

## **References**

- [1] BOLLOBÁS, B.: *Modern Graph Theory*. Springer–Verlag
- [2] GIBBONS, A.: *Algorithmic Graph Theory*. Cambridge University Press
- [3] Golumbic, Martin (1980), *Algorithmic Graph Theory and Perfect Graphs*, Academic Press.
- [4] Harary, F. *Graph Theory*. Reading, MA: Addison-Wesley, 1994.
- [5] Harary, Frank (1969), *Graph Theory*, Reading, MA: Addison-Wesley.
- [6] Harary, Frank; Palmer, Edgar M. (1973), *Graphical Enumeration*, New York, NY: Academic Press
- [7] MCHUGH, J.A.: *Algorithmic Graph Theory*. Prentice–Hall